



9
D. Bond
11/26/01

RECEIVED

NOV 16 2001

Group 2100

219.36435X00

Serial No. 09/215,788

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF APPEALS AND INTERFERENCES**

Appellants: Jerrie L. COFFMAN et al.

Application No.: 09/215,788

Filing Date: December 21, 1998

Title: EFFICIENTLY EXPORTING LOCAL DEVICE ACCESS ONTO A
SYSTEM AREA NETWORK USING A DIRECT-CALL
INTERFACE

Art Unit: 2152

Examiner: B. Prieto

APPEAL BRIEF

Assistant Commissioner of Patents
Washington, D.C. 20231

November 13, 2001

Sir:

Pursuant to Appellants' earlier filed Notice of Appeal dated on August 10, 2001, Appellants hereby appeal to the Board of Appeals and Interferences from the decision (Paper No. 5) dated on June 15, 2001 with respect to all the finally rejected claims 1-28.

This Appeal Brief is being filed in triplicate.

11/14/2001 EDENB021 00000021 09215788

01 FC:120

320.00 OP

219.36435X00
Serial No. 09/215,788

I. Statement of Real Party in Interest

Pursuant to 37 C.F.R. §1.92(c)(1)(as amended), the real party in interest is:

Intel Corporation
2200 Mission College Blvd., SC4-202
Santa Clara, CA 95052

II. Related Appeals and Interferences

Pursuant to 37 C.F.R. §1.192(c)(2)(as amended), although the real party in interest has other pending appeals and interferences, none of the other pending appeals and interferences is believed to directly affect or be directly affected by, or to have any bearing upon the decision of the Board of Patent Appeals and Interferences in this appeal.

III. Status of the Claims

Claims 1-28 are pending in this application at the filing of this Brief. Claims 1-28 have been rejected under 35 U.S.C. §103(a) as being unpatentable over Heil et al., U.S. Patent No. 6,173,374, as modified to incorporate selected features from Angelo et al., U.S. Patent No. 6,6061,794. Appellants are appealing from the outstanding rejection of claims 1-28. Of the appealed claims, claims 1, 7, 14 and 22-23 are independent claims.

IV. Status of the Amendments

An Amendment after final under 37 C.F.R. §1.116(b) was filed on August 10, 2001,

and will be entered upon filing of this Appeal Brief.

V. Summary of the Invention

One of inherent challenges of a system area network (SAN) cluster as shown in FIG. 1 which links remote servers 10-18 and network-connected storage devices 130is to design a data transport mechanism that can deliver and exchange a large amount of data messages quickly between end nodes in the SAN cluster.

Traditional data transports between end nodes (for example, remote servers 10-18) in a SAN cluster are done through the network infrastructure provided by a host operating system (OS). However, a large amount of system processing overhead and an extended processing time are required to process each data message between the host server 10 and remote servers 14-18 as shown in FIG. 2. This overhead limits input/output (I/O) bandwidth, increases input/output (I/O) latency, and increases the response time to the application.¹

Another data transport mechanism between end nodes is to use another unique application to generate special application-to-application messages to the remote node in order to access a remote server in a SAN cluster. A remote application running on the remote node in the SAN cluster must issue an input/output (I/O) request to the remote operating system (OS) on behalf of the local application. This way the operating system (OS) overhead

¹ See page 3, lines 18-23, and page 4, lines 1-2 of Appellants' specification.

on the local server (host server) may be avoided. However, there are still a great deal of undesirable coordination between cooperating applications of the local server and the remote server in the SAN cluster.²

The present invention seeks a more compact driver system solution for a host system (server) to provide a direct, transparent access to I/O storage devices connected to the host server within a SAN for efficient sharing of resources and databases among all network members without incurring the overhead of the operating system (OS) protocol stack and without coordinating special application-to-application messages between nodes in a system area network (SAN) cluster.³

FIG. 3 illustrates a host driver module 310 installed as part of an operating system (OS) of the host server 300 and dynamically configured to accept communication requests for I/O device access from remote servers 340 and 350, via a SAN network interface card (NIC) 328 according to an embodiment of the present invention. Such a host driver module 310 ("IOP" access module) may be provided in a form of a hardware module, a combined hardware/software module, or a software module which can be easily downloaded (e.g., from a tangible medium such as a disk, or via the Internet) into the host operating system (OS) for allowing I/O storage devices 326 to operate independently from both the specific device

² See page 4, lines 12-18 of Appellants' specification.

³ See page 8, lines 10-15 of Appellants' specification.

controlled and the host operating system (OS).⁴

In addition to the host driver module 310, an Input/Output Platform (IOP) 320 is also arranged, via a PCI bus 318, for controlling an array of disk storage devices 326, including transporting data from the host driver module 310 to the local IOP 320 and, vice versa.

The IOP 320 contains a device driver module 322 which resides on and interfaces with the particular controller and I/O storage devices 326, and a communication layer 324 which defines an open, standard mechanism for communication between the host driver module 310 and the device driver module 322. The device driver module 322 is responsible for control and data transfer of I/O storage devices 326. The communication layer 324 may include a message layer which sets up a communication session, and a transport layer which defines how information will be shared. Collectively, the communication layer 324 may be responsible for managing all requests, and providing a set of Application Programming Interfaces (APIs) for delivering messages, along with a set of support routines that process the messages.⁵

The host driver module 310, as shown in FIG. 3, comprises a Local Transport 314 arranged to provide an interface to the IOP 320 supporting an array of I/O storage devices 326; a Remote Transport 316 arranged to provide an interface to another remote system such

⁴ See page 8, lines 18-21 of Appellants' specification.

⁵ See page 9, lines 13-22 of Appellants' specification.

as remote servers 340 and 350, via the SAN 330; and a Connection Manager 312 arranged to establish connection services and to create a direct call path between the Local Transport 314 and the Remote Transport 316 so as to provide direct access to I/O storage devices 326 without incurring the overhead of the operating system (OS) protocol stack and without coordinating special application-to-application messages.⁶

When the host driver module 310 is initialized, the Local Transport 314 scans the PCI bus 318 to locate and initialize all local IOPs 320 and builds an opaque "context" structure for each IOP found. The Remote Transport 316 prepares to accept requests from a remote server 340 or 350 through the SAN network interface card (NIC) 328. The Connection Manager 312 then queries the Local Transport 314 to determine the number of IOPs and builds a descriptor structure for each IOP. The IOP descriptor structure, as shown in FIG. 4, includes an exported table of function call pointers and the context required by the Local Transport 314 to communicate with each IOP. Next, the Connection Manager 312 establishes a SAN management communication channel through the Remote Transport 316, and waits for an external connection from a remote server 340 or 350 on a SAN 330.⁷

FIG. 4 illustrates an example IOP descriptor structure established during initialization for a direct call interface between Local Transport 314 and Remote Transport 316 by the

⁶ See page 10, lines 12-21 of Appellants' specification.

⁷ See page 11, lines 3-18 of Appellants' specification.

Connection Manager 312 of the host driver module 310 for an inbound message from a remote SAN server for direct access to local IOP 320 according to an embodiment of the present invention. The Local Transport 314 builds a list of different IOP context structures if different IOPs are available in the host server 300. Each IOP context structure includes different designations for direct access to each IOPs. In addition, the Local Transport 314 also has a send handler function which is a program interface to receive an inbound message from a remote SAN server for direct access to local IOP 320.⁸

The Connection Manager 312 builds an IOP descriptor structure for each IOP found. Each IOP descriptor structure includes an exported table of function call pointers such as IOP context pointer and send handler function pointer required by the Local Transport 314 to communicate with the IOP 320. The Remote Transport 316 builds an IOP connection structure including at least an IOP descriptor pointer which refers to the IOP descriptor structure of the Connection Manager 312 for making a direct call to the Local Transport 314 through the send handler function. In addition, the Remote Transport 316 also has a receive handler function which is a program interface to receive an inbound message from a remote server on a SAN for direct access to local IOP 320 and to deliver an outbound message to a remote server on a SAN. For an outbound message to a remote server on a SAN, data structures established for a direct call interface between Local Transport 314 and Remote

⁸

See page 11, lines 19-22 and page 12, lines 1-8 of Appellants' specification.

Transport 316 by the Connection Manager 312 as shown in FIG. 4 remain the same. However, the communication directions between the send handler function in the Local Transport 314 and the receive handler function in the Remote Transport 316 are reversed.⁹

After initialization, a direct call interface between the Local Transport 314 and the Remote Transport 316 of the host driver module 310 is established to provide a direct, transparent access to I/O storage devices 326 within a system area network (SAN) cluster 330 that bypasses the traditional OS protocol stacks for efficient sharing of resources without incurring the overhead of the OS stack and coordinating special application-to-application messages between nodes of a SAN cluster 330.

First, in order to establish a service connection to an IOP 320, that is a logical connection of an IOP 320 to a remote server 340 or 350 for the purpose of sending messages and transporting data therebetween, the remote server 340 or 350 in a system area network (SAN) 330 exchanges messages with the Connection Manager 312 of the host drive module 310 based on a protocol specified by the network used and/or the Intelligent I/O Architecture. The Connection Manager 312 next advertises the presence of local IOPs that are available for external use.¹⁰

⁹ See page 12, lines 8-21 of Appellants' specification.

¹⁰ See page 13, lines 10-19 of Appellants' specification.

When a remote server 340 or 350 requests a service connection to an IOP 320 of the host server 300, the Connection Manager 312 passes the address of an IOP descriptor structure as shown in FIG. 4 to the Remote Transport 316 to establish the service connection. The IOP descriptor structure provides the Remote Transport 316 access to the function call pointers and context required by the Local Transport 314 to communicate directly with the IOP 320.

When a service connection to a local IOP 320 is established by remote server 340 or 350 and the Connection Manager 312 of the host server 300, low latency and high-bandwidth messages passing between nodes is obtained by bypassing the layers of OS protocol stacks when sending and receiving messages. Data structure pointers are exchanged to establish a direct call relationship between software modules on the host server 300 within a system area network (SAN) 330. To deliver an inbound message from a remote server on a system area network (SAN) 330, the receive handler function in the Remote Transport 314 simply refers to the IOP descriptor structure of the Connection Manager 312 to make a direct call to the send handler function in the Local Transport 314, providing the function with the IOP context and the message frame pointer. Likewise, an outbound message from the Local Transport 314 is delivered to a send handler function (not shown) in the Remote Transport 316. However, the outbound message includes a pointer to a structure containing the function address and context required by the Remote Transport 316 to send the message to a remote

server 340 or 350 within a system area network (SAN) 330.¹¹

FIG. 5 illustrates an example step-by-step process of routing an inbound message or data from a remote server 340 or 350 of a system area network (SAN) through the Local Transport 314 and Remote Transport 316 by the Connection Manager 312 of a host driver module 310 according to an embodiment of the present invention. Each message may be transmitted in a series of frames. When a service connection is requested from a remote server, an inbound message frame from a remote server on the system area network (SAN) is delivered to the receiver handler function in the Remote Transport 316 at step 510. The address of the IOP connection structure describing the service connection on which the inbound message was received (i.e., the context of the service connection) is located at the Remote Transport 316 at step 520. The IOP descriptor structure is located by using an IOP descriptor pointer to refer to the IOP descriptor structure of the Connection Manager 312.

Next, a context field in the inbound message frame is saved and replaced with a new context field for the Remote Transport 316 at step 530. The pointer to the IOP descriptor structure of the Connection Manager 312 is retrieved from the IOP connection structure at step 540. Then, the send handler function pointer and the IOP context pointer are read from the IOP descriptor structure of the Connection Manager 312 at step 550. The send handler function in the Local Transport 314 is called in order to pass the inbound message frame

¹¹

See page 14, lines 3-16 of Appellants' specification.

using the IOP context pointer in step 560.¹²

Likewise, FIG. 6 illustrates an example step-by-step process of routing an outbound message or data from a local I/O platform (IOP) 320 of a host server 300 to a remote server 340 or 350, via a SAN 330 according to an embodiment of the present invention. Again, each message is transmitted in a series of message frames. When an outbound message is routed to a remote server 340 or 350 on a system area network (SAN) 330, an outbound message frame is delivered to the send handler function in the Local Transport 314 at step 610. The context field in the outbound message frame is read at step 620. This context is a pointer to a callback object structure in the Remote Transport 316 which contains the address of the Remote Transport IOP connection structure for a service connection.

Next, the callback function is called, passing the callback context pointer and the outbound message frame as outgoing parameters at step 630. At the Remote Transport 316, an outgoing message frame and the IOP connection structure address are delivered to the send handler function at step 640. A context field in the outgoing message frame is replaced with the saved context from the original inbound message frame as described with reference to FIG. 5, at step 650. Then, the outgoing message frame is sent out the service connection to a remote server 340 or 350 on a SAN 330 at step 660.¹³

¹² See page 15, lines 1-12 of Appellants' specification.

¹³ See page 16, lines 1-7 of Appellants' specification.

219.36435X00
Serial No. 09/215,788

As a result, the host driver system according to an embodiment of the present invention can export direct I/O storage device access without incurring the overhead of the traditional network infrastructure in the operating system (OS), and without using special application-to-application messages between nodes within a SAN cluster.

VI. Issues

Whether claims 1-28 are unpatentable under 35 U.S.C. §103(a) as rendered obvious over a proposed combination of Heil et al., U.S. Patent No. 6,173,374, and Angelo et al., U.S. Patent No. 6,6061,794.

VII. Grouping of Claims

Independent claims 1, 7, 14 and 22-23 are argued separately, with the arguments necessarily carrying forward to their respective dependent claims 2-6, 8-13, 15-21 and 24-28. In addition, each of dependent claims 2-6, 8-13, 15-21 and 24-28 is also argued separately. Therefore, for purposes of the rejections under 35 U.S.C. §102(e), claims 1-28 stand or fall independently of each other under 37 C.F.R. §1.192(c)(5) for the reasons set forth in the arguments hereinbelow.

VIII. Arguments

Claims 1-28 are deemed patentable over the proposed combination of Heil et al., U.S. Patent No. 6,173,374 and Angelo et al., U.S. Patent No. 6,061,794.

Claims 1-28 have finally been rejected under 35 U.S.C. §103(e) as being unpatentable over the Examiner's proposed combination of Heil et al., U.S. Patent No. 6,173,374 (hereinafter referred as Heil '374) and Angelo et al., U.S. Patent No. 6,061,794 (hereinafter referred as Angelo '794). On pages 1-8 of the final Office action (Paper No. 5) dated on June 15, 2001, the Examiner copies each of Appellants' claims 1-28 *verbatim* and then flip-flops between what is disclosed by Heil '374 and Angelo '794 to support an assertion that the proposed combination of Heil '374 and Angelo '794 somehow discloses all of the limitations of Appellants' claims 1-28 as pending on appeal.

For example, the Examiner asserts, in support of the rejection of independent claim 23, that Heil '374 as a primary reference discloses:

an access module including a processor(s) associated memory, and a local memory bus, and an input/output bus forming an input/output platform (IOP) access module (Fig. 1-5A, col 6/lines 34-64), module for providing input/output device access between a host system and another system (col 3/lines 64-col 4/line 29), module arranged to establish a service connection to a local (IOP) connected to a local bus using a driver module in response to a request from a remote server on a computer system network, establishing connection process comprising the steps of: beginning initialization of said driver module (col 12/lines 26-37, Fig. 4A-4D, steps 500, 510, 518, 529) which provides access to a local storage system while bypassing protocol stacks of a host operating system (providing a direct request call access while bypassing OS specific portion layer, bypassing 250, 260 access means independent of OS, Fig. 2, col 10/lines 28-65), said driver module comprising a module 180 (Local Transport) which provides direct access to the local storage device system (col 4/lines 3-20), a module 181 (Remote Transport) which interfaces to other nodes of said system network (col 4/lines 21-29, Fig. 5A remote node N), and connection means (Connection Manager) which provides connection services and coordinating functions responsible for creating a direct call path between the Local Transport and the Remote

Transport (direct path between the module 118 and the module 181 so as to provide access to input/output (117.8 and 117.9) devices driver coupled to local storage (Fig. 5A, 118) devices); searching process (scanning) to locate and initialize all local input/output platforms (IOPs) peer HBA modules, and building an IOP context directory structure (map) (i.e. descriptors and addresses of routines located with a region of memory) for each input/output platform (IOP) found responsive (col 12/lines 9-col 13/line 3); (managing, making arrangements for, i.e. preparing) means to take, be given, receive (i.e. accept) a request for a service connection from said remote server on said system network (col 11/lines 36-53, col 12/lines 9-19, Fig. 3); query via scanning means itemize responsive input/output IOP on build map (col 11/lines 53-col 65, col 12/lines 8-59), and building a descriptor structure for each IOP which includes an exported directory table having function call (col 13/lines 5-15) address pointer and characteristic parameters to support communication with the input/output platform (IOP) found (col 12/lines 9-col 13/line 13); and establishing a communication channel through the Remote Transport, which waits for an external connection from said remote server on said system network for shipping (exporting) local device access resources onto said system network using said direct call path between the Local Transport and the Remote Transport (col 10/lines 28-65). See pages 1-2 of Paper No. 5.

The Examiner has expressly admitted on the same page of Paper No. 5 that,

Heil ['374] does **not** teach an access module arranged to establish a service connection to a local input/output platform (IOP) running on host server of particular system area network and establishing a associated system area network management communication channel; nor module (180) arranged to provide an interface to an input/output platform (IOP) supporting an array of input/output devices is denoted "Local Transport", nor module (181) arranged to provide an interface to said another system is denoted "Remote Transport". See page 2, Paper No. 5.

The Examiner then cites col. 4, lines 21-32 and col. 5, lines 1-40 of Angelo '794 for allegedly disclosing these features in order to enable an artisan to modify the host system of Heil '374 to incorporate means provided by Angelo '794 to arrive at Appellants' disclosed invention as

defined, for example, in claim 23.

However, when confronted with evidence presented by Appellants that no where in Angelo '794 is there disclosure of any driver module denoted as "Local Transport", "Remote Transport" and "Connection Manager" as identified by the Examiner as lacking from Heil '374, the Examiner ignores Angelo '794 completely and argues in the same final Office Action (Paper No. 5) that Heil '374 now also discloses,

an access module, module including a processor(s) (100) associated memory (105), and a local memory bus (PCI 116.5, i.e., I/O bus), an interface input/output processor (117), element(s) (117 alone or with 116) form an input/output platform IOP access module (Heil: Fig. 1-5A, host system, col 6/lines 34-64, host 150 communicatively coupled to host 151 via network medium, col 8/lines 12-23), module for providing input/output device access between a host (151) system and another (151) system (Heil: I/O means, col 3/lines 64-col 4/line 29) comprising: a module (Heil: Fig. 5A, 180) arranged to provide interface (Heil: Fig. 1, 117.8, 117.9) means to an input/output platform (IOP) supporting an array of input/output (118) storage devices (Heil: col 4/lines 3 -20); a module (Heil: Fig. 5A, 181) arranged to provide an interface (Heil: Fig. 5A, 120) means to said another system node (Heil: Fig 5A and 1, col 4/lines 21-29); and module comprising (Heil: Fig. 5A, 171) arranged to establish connection services and to create a direct call path between the module 118 and the module 181 so as to provide access to input/output (117.8 and 117.9) devices driver coupled to local storage (Heil: Fig. 5A, 118) devices (direct call path means established via a peer-to-peer communication path (i.e., devices on a layered communication network that operate on the same application protocol level), direct peer-to-peer communication path, col 8/lines 8-24, nodes having direct interface-to-interface connection via communication medium, col 8/lines 50-56, direct port-to-port connection via device drives, col 10/lines 28-50.

Therefore Heil teachings ... host system which comprises a module arranged to provide an interface to an input/output platform (IOP) supporting an array of input/output devices; and a module arranged to provide an interface to said another system; and module arranged to established [sic] connection services

and to create direct call path between the said [sic] module providing an interface to an input/output platform (IOP) supporting an array of input/output devices and said module providing an interface to said another system so as to provide access to said input/output devices. See page 10 of Paper No. 5 dated on June 15, 2001.

In short, Heil '374 as a primary reference discloses all features of Appellants' independent claims 1, 7, 14 and 22-23, and Angelo '794 as a secondary reference is not being relied upon for any reasons. Notwithstanding the Examiner's comments, the §103 rejection of Appellants' claims 1-28 is both factually and legally incorrect because neither Heil '374 nor Angelo '794, whether taken individually or in combination, discloses the claimed features as the Examiner has alleged. Virtually all the citations from either Heil '374 or Angelo '794 are misplaced. Even more puzzling is the fact that no where in Angelo '794 as a secondary reference is there disclosure of any module denoted as "Local Transport", "Remote Transport" and "Connection Manager" as part of a host driver module as identified by the Examiner as lacking from Heil '374. Therefore the rejection of Appellants' claims 1-28 is respectfully traversed for reasons as discussed hereinbelow.

A. The Examiner failed to establish a *prima facie* case of obviousness of Appellants' independent claims 1, 7, 14 and 22-23 because there is no factual evidence to support such a conclusion of obviousness.

In rejecting claims under 35 U.S.C. §103, the Examiner bears the initial burden of establishing a *prima facie* of obviousness. In re Rijckaert, 9 F.3d 1531, 1532, 26 USPQ2d

1955, 1956 (Fed. Cir. 1993); In re Oetker, 977 F.2d 1443, 1445 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). Only if this burden is met does the burden of coming forward with rebuttal argument or evidence shift to the Appellants. Rijckaert, 9 F.3d at 1532, 26 USPQ2d at 1956. When the references cited by the Examiner fail to establish a *prima facie* case of obviousness, the rejection is improper and must be overturned. In re Fine, 873 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). Obviousness under 35 U.S.C. §103 is a legal conclusion based on factual evidence, not a factual determination. Graham v. Deere, 383 U.S. 1, 148 USPQ 459. It is incumbent upon the Examiner to establish a factual basis to support the legal conclusion of obviousness. In re Fine, 873 F.2d at 1074, 5 USPQ2d at 1598. Determination of obviousness must be based on facts, and not on unsupported generalities. In re Warner, 379 F.2d 1011, 154 USPQ 173 (CCPA 1967); In re Freed, 425 F.2d 785, 165 USPQ 570 (CCPA 1970). Any deficiencies in the factual basis cannot be supplied by resorting to speculation or unsupported generalities. Id.

In the present situation, the Examiner has **not** met the initial burden of producing factual evidence to establish a *prima facie* case of obviousness. Each of Appellants' independent claims 1, 7, 14 and 22-23 defines a host driver module installed in a host system comprising a "Local Transport arranged to provide an interface to the IOP supporting an array of I/O storage devices"; a "Remote Transport arranged to provide an interface to another remote system, via the SAN"; and a "Connection Manager arranged to establish connection services and to create a direct call path between the Local Transport and the

Remote Transport so as to provide direct access to I/O storage devices.”

For example, independent claims 1, 7, 14 and 22 define a host driver module (IOP access module) installed in a host server for exporting I/O storage device access onto a computer network which comprises:

a Local Transport arranged to provide an interface to an input/output platform (IOP) supporting an array of input/output devices;

a Remote Transport arranged to provide an interface to said another system, via said data network; and

a Connection Manager arranged to establish connection services and to create a direct call path between the Local Transport and the Remote Transport so as to provide access to input/output devices.

As expressly defined in each of Appellants' independent claims 1, 7, 14 and 22 the host driver module 310 installed in a host server including a Connection Manager 312, a Local Transport 314 and a Remote Transport 316 for the purposes of exporting device access to remote devices on a data network (SAN) as shown in FIG. 3. For example, Local Transport 314 is arranged to provide an interface to IOP 320 on the PCI bus 318 supporting an array of IO devices 326. Remote Transport 316 is arranged to provide an interface to remote devices (remote servers) via a SAN. Connection Manager 312 is utilized to establish connection services and create a direct call path between the Local Transport 314 and the Remote Transport 316 so as to provide access to IO devices 326. Data structure pointers as shown in FIG. 4 are exchanged between the Local Transport 314 and the Remote Transport

316 by way of the Connection Manager 312 in order to establish a direct-call relationship between separately installed software modules in order to avoid incurring the overhead of the operating system (OS) protocol stack and coordinating special application-to-application messages as required by the SAN cluster.

The host driver module 310 may be initialized as defined by independent process claim 23 as follows:

beginning initialization of said driver module which provides access to a local storage system while bypassing protocol stacks of a host operating system, said system driver module comprising a Local Transport which provides direct access to the local storage device system, a Remote Transport which interfaces to other nodes of said system area network, and a Connection Manager which provides connection services and coordinates functions responsible for creating a direct call path between the Local Transport and the Remote Transport;

scanning, at said Local Transport, the local bus to locate and initialize all local input/output platforms (IOPs), and building an IOP context structure for each input/output platform (IOP) found;

preparing, at said Remote Transport, to accept a request for a service connection from said remote server on said system area network;

asking, at said Connection Manager, whether said Local Transport determines the number of input/output platforms (IOPs), and building a descriptor structure for each input/output platform (IOP) which includes an exported table of function call pointers and the context required by the Local Transport to communicate with the input/output platform (IOP); and

establishing a system area network management communication channel through the Remote Transport, which waits for an external connection from said remote server on said system area network for exporting local device access onto said system area network using said direct call path between the Local Transport and the Remote Transport.

In other words, the Local Transport 314 scans the PCI bus 318 to locate and initialize all local IOPs and builds an opaque “context” structure for each IOP found. The Remote Transport 316 then prepares to accept requests through network interface card (NIC) 328. The Connection Manager 312 then queries the Local Transport 314 to determine the number of IOPs and builds a descriptor structure for each IOP (IOP descriptor structure includes an exported table of function call pointers and the context required by the Local Transport 314 to communicate with the IOP), and establish a management communication channel through the Remote Transport 316, which waits for an external connection from a remote server via a SAN for exporting local device access using said direct call path between the Local Transport 314 and the Remote Transport 316.

The host driver module 310 allows, for example, a distributed database running on a cluster of servers to share and directly access all the storage in the cluster transparently. As a result, the overhead incurred by the OS stack on the remote node is avoided via “short-circuiting” at the driver level, and no unique database is required to generate special application-to-application messages to remote nodes in order to access IO storage devices located on remote storage.

In contrast to Appellants’ independent claims 1, 7, 14 and 22-23, Heil ‘374 as a primary reference discloses a system for I/O shipping of requests using peer-to-peer communications between host bus adapters ("HBAs") connected to I/O devices.

Specifically, Heil ‘374 discloses the use of one or more host bus adapters HBA 112 as

shown in FIG. 2 and elements 180, 181 as shown in FIG. 5A installed in a host system of a clustered computer network for processing IO requests received from the host system. Each HBA also serves as a network interface card (NIC) connected to a peer HBA via a Fibre Channel backbone 121 (high-speed communication medium) and contains therein a directory within memory 116 for storing location information regarding blocks of data stored in storage devices and software for searching the directory to determine whether to locally or remotely retrieve blocks of data. The HBA installed in one node of a clustered computer network is operable to establish and maintain communications with at least one other HBA installed in another node of the clustered computer network. However, the HBA of Heil '374 is NOT and does not correspond to a host driver module as defined by Appellants' independent claims 1, 7, 14 and 22-23.

There is **no** disclosure anywhere from Heil '374 of Appellants' claimed "host driver module" installed in a host system which comprises "a Local Transport arranged to provide an interface to an input/output platform (IOP) supporting an array of input/output devices;" "a Remote Transport arranged to provide an interface to said another system;" and "a Connection Manager arranged to establish connection services and to create a direct call path between the Local Transport and the Remote Transport so as to provide access to IO devices" as generally defined in Appellants' independent claims 1, 7, 14 and 22-23.

Nevertheless, the Examiner argues that Heil '374 discloses:

an access module, module including a processor(s) (100) associated memory

(105), and a local memory bus (PCI 116.5, i.e., I/O bus), an interface input/output processor (117), element(s) (117 alone or with 116) form an input/output platform IOP access module (Heil: Fig. 1-5A, host system, col 6/lines 34-64, host 150 communicatively coupled to host 151 via network medium, col 8/lines 12-23), module for providing input/output device access between a host (151) system and another (151) system (Heil: I/O means, col 3/lines 64-col 4/line 29) comprising: a module (Heil: Fig. 5A, 180) arranged to provide interface (Heil: Fig. 1, 117.8, 117.9) means to an input/output platform (IOP) supporting an array of input/output (118) storage devices (Heil: col 4/lines 3 -20); a module (Heil: Fig. 5A, 181) arranged to provide an interface (Heil: Fig. 5A, 120) means to said another system node (Heil: Fig 5A and 1, col 4/lines 21-29); and module comprising (Heil: Fig. 5A, 171) arranged to establish connection services and to create a direct call path between the module 118 and the module 181 so as to provide access to input/output (117.8 and 117.9) devices driver coupled to local storage (Heil: Fig. 5A, 118) devices (direct call path means established via a peer-to-peer communication path (i.e., devices on a layered communication network that operate on the same application protocol level), direct peer-to-peer communication path, col 8/lines 8-24, nodes having direct interface-to-interface connection via communication medium, col 8/lines 50-56, direct port-to-port connection via device drives, col 10/lines 28-50.

However, no where in Heil '347 is there disclosure of any host driver module including modules denoted as "Local Transport", "Remote Transport" and "Connection Manager" as identified by the Examiner. In fact, all cited portions of Heil '347 are misplaced.

For example, the cited FIGs. 1-5A, col 6/lines 34-64 and col 3/lines64-col 4/line 29 of Heil '347 for allegedly disclosing "[an input/output platform (IOP) access] module for providing input/output device access between a host (151) system and another (151) system" [preamble of claims 1, 7, 14 22-23] is incorrect. Rather, that cited portion of Heil '347 refer

to the use of a host bus adapter "HBA" which is a device which adapts (connects) a host computer system as shown in FIG. 1 to an I/O device as well as to connect to a Fibre Channel backbone. The host bus adapter "HBA" 117 of Heil '347 may correspond to a network interface card "NIC" 328 shown in FIG. 3 of Appellants' disclosed invention which is arranged to adapt a host computer system to an I/O device as well as to connect to a SAN. However, the HBA 117 of Heil '347 does not constitute a host driver module (IOP access module) as defined in Appellants' independent claims 1, 7, 14 and 22-23, and shown in FIG. 3 as a separate module from the SAN NIC 328.

Likewise, the cited FIGs. 1 and 5A, 181, 120, 171, 117.8, 117.9, and col. 4/lines 21-29 of Heil '374 for allegedly disclosing Appellants' claimed "host driver module including a Local Transport, a Remote Transport and a Connection Manager" is also incorrect.

FIG. 5A, element 181 of Heil '374 refers to a host bus adapter "HBA". FIG. 5A, element 120 of Heil '374 refers to a fiber channel chip which is a critical component of the HBA that is connected to the PCI bus 116.5 and the Fibre Channel backbone 121. FIG. 5A, element 171 of Heil '374 refers to a front-end interface (IF) of either a HBA 180 or HBA 181 to provide an interface to a PCI bus 116.5. Neither of these cited elements constitutes the features of Appellants' claimed host driver module including a Local Transport, a Remote Transport and a Connection Manager as expressly defined in each of independent claims 1, 7, 14 and 22-23.

In fact, as shown in FIG. 5A (an alternative embodiment) of Heil '374, two HBAs 180

and 181 may be utilized in a single node (or host computer system). HBA 180 is used to process IO requests directed to local devices 118. HBA 181 is used to support remote IO requests from the Fibre Channel backbone 121. These two HBA 180 and 181 are hardware devices configured to perform stated specific functions. The Examiner simply cannot cite a single HBA 117 (see FIG 1) in one embodiment of Heil '374 to correspond Appellants' claimed single host driver module (IOP access module), and cite two other HBAs 180, 181 (see FIG. 5) in another embodiment of Heil '374 to corresponding the Local Transport and Remote Transport components of Appellants' claimed host driver module (IOP access module), and then cite an internal hardware of the same HBAs 180, 181 to correspond the Connection Manager component of Appellants' claimed IOP access module. This is highly improper. Two prior art HBA devices cannot be distorted and improperly interpreted in such a way just to read on Appellants' claimed host driver module (IOP access module), particularly when the prior art HBA hardware devices are completely different from Appellants' claimed host driver module (IOP access module).

Similarly, the cited col. 6, lines 34-64 of Heil '374 for allegedly disclosing Appellants' claimed "service connection to a local input/output platform (IOP) connected to a local bus using a driver module" is also incorrect. That cited portion of Heil '374 simply depicts a host system as shown in FIG. 1 including essential components such as CPU 100, cache 105, processor bus 110, host-to-PCI bus bridge 115 and a PCI bus 116. No reference to any input/output platform (IOP) as defined in claim 23, see FIG. 3.

The cited col. 4, lines 3-20 of Heil '374 for allegedly disclosing Appellants' claimed "module (Local Transport) which provides direct access to the local storage system" is also incorrect. That cited portion of Heil '374 describes a host bus adapter (HBA) which is a hardware component adapted to connect a host system to an IO device as well as to connect to a Fibre Channel backbone. The HBA of Heil '372 is analogous to the SAN network interface card (NIC) 328 shown in FIG. 3 of Appellants' disclosed invention.

In contrast to the HBA of Heil '372, Appellants' claimed "Local Transport" is a part of the host driver module (IOP access module) along with the Remote Transport and the Connection Manager and is arranged to provide an interface to an IOP supporting an array of IO devices.

The cited col. 4, lines 21-29, FIG. 5A of Heil '374 for allegedly disclosing Appellants' claimed "module (Remote Transport) which interfaces with other nodes of said system network" is also incorrect. That cited portion of Heil '374 describes how the host bus adapter (HBA) which is a hardware component installed in each node of a Fibre Channel is able communicate among each other as peers. Again, **no** disclosure of any "Remote Transport".

The cited col. 12, line 9 extending to col. 13, line 13 of Heil '374 for allegedly disclosing Appellants' claimed "building [an IOP] descriptor structure for each input/output platform (IOP) which includes an exported table of function call pointers and the context required by the Local Transport to communicate with the input/output platform (IOP)" is

likewise incorrect. That cited portion of Heil '374 only describes how the host system and the HBA are initialized and how the HBAs build a local directory containing the mapped location information for respective local storage subsystems 504.

The Examiner argues that Appellants' claimed IOP descriptor structure as shown in FIG. 4 is analogous to the use of an HBA directory (map) (i.e. descriptors and addresses of routines located with a region of memory). However, this line of argument is incorrect. This is because Appellants' claimed "IOP descriptor structure" is established for a direct call interface between the Local Transport 314 and Remote Transport 316 in order to access to each IOP, whereas the HBA directory of Heil '374 is used to map location information of local storage subsystems. Therefore, no IOP descriptor structure is disclosed by Heil '374.

Lastly, the cited col. 6, lines 34-64 of Heil '374 for allegedly disclosing the "service connection to a local input/output platform (IOP) connected to a local bus using a driver module" is also incorrect. Rather, that cited portion of Heil '374 simply depicts a host system as shown in FIG. 1 including essential components such as CPU 100, cache 105, processor bus 110, host-to-PCI bus bridge 115 and a PCI bus 116. Similarly, the cited col. 4, lines 3-20 of Heil '374 does NOT disclose any claimed "module (Local Transport) which provides direct access to the local storage system". Rather, that cited portion of Heil '374 only describes a host bus adapter (HBA) which is a hardware component adapted to connect a host system to an IO device as well as to connect to a Fibre Channel backbone. Likewise, the cited col. 4, lines 21-29, FIG. 5A of Heil '374 does not disclose any claimed "module

(Remote Transport) which interfaces with other nodes of said system network". Instead, that cited portion of Heil '374 only describes how the host bus adapter (HBA) which is a hardware component installed in each node of a Fibre Channel is able communicate among each other as peers. Furthermore, the cited FIG. 5A, element 171 of Heil '374 does not disclose any claimed "Connection Manager which provides connection services and coordinate functions responsible for creating a **direct call path between the Local Transport and the Remote Transport**" as defined in Appellants' claims 1, 7, 14 and 22-23. Rather, element 171 is simply a front-end interface as shown in FIG. 5A for providing an interface to a PCI bus 116.5. There is no disclosure of any "Connection Manager" whatsoever.

In view of the foregoing reasons and the complete failure of Heil '374 and Angelo '974 to disclose the claimed features as alleged by the Examiner, Appellants respectfully request that the rejection of Appellants' independent claims 1, 7, 14 and 22-23 be reversed.

B. The Examiner erred in making the modification of Heil '374 to incorporate selected features from Angelo '974 to arrive at Appellants' independent claims 1, 7, 14 and 22-23 because there is no suggestion in either Heil '374 or Angelo '974 to support such a modification.

The mere fact that a prior art device could be modified to produce the claimed invention does not justify an obviousness rejection unless the prior art suggested the desirability of the modification (emphasis in original).

In re Laskowski, 871 F.2d 115, 10 USPQ2d 1397 (Fed. Cir. 1999); quoting In re Gordon, 733 F.2d 900, 902, 221 USPQ 1125, 1127 (Fed. Cir. 1984). In re Fine, 873 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). In re Freed, 425 F.2d 785, 165 USPQ 570 (CCPA 1970). Therefore, even assuming *arguendo* that Angelo '974 discloses a host driver module installed in a host system comprising a "Local Transport arranged to provide an interface to the IOP supporting an array of I/O storage devices"; a "Remote Transport arranged to provide an interface to another remote system, via the SAN"; and a "Connection Manager arranged to establish connection services and to create a direct call path between the Local Transport and the Remote Transport so as to provide direct access to I/O storage devices" as defined in independent claims 1, 7 , 14 and 22-23, which Appellants do not believe, there is still no reason for an artisan to make the modification in the manner suggested by the Examiner in order to arrive at Appellants' claimed invention.

In the present situation, neither Heil '374 nor Angelo '974 suggests any implementation of a host driver module including a Local Transport, a Remote Transport and a Connection Manager for providing direct access to I/O storage devices. In view of the complete failure of Heil '374 and Angelo '974 to suggest the modification, Appellants respectfully submit that the proposed combination of Heil '374 and Angelo '974 does not make obvious Appellants' claimed invention. Accordingly, the rejection of Appellants' independent claims 1, 7, 14 and 22-23 should be reversed.

C. Dependent claims 2-6, 8-13, 15-21 and 24-28 are deemed separately patentable over Heil '374, as modified to incorporate Angelo '794.

Claims 2-6, 8-13, 15-21 and 24-28 which depend from claims 1, 7, 14 and 22-23 are deemed patentable from claims 1, 7, 14 and 22-23 if their parent claims 1, 7, 14 and 22-23 are patentable. Hartness Int'l, Inc., v. Simplicatic Eng'g Co., 891 F.2d 1100, 1108, 2 USPQ2d 1826, 1831 (Fed. Cir. 1987); In re Abele, 684 F.2d 909, 214 USPQ 682, 689 (CCPA 1982) *see also* In re Sernaker, 702 F.2d 989, 991, 217 USPQ 1, 3 (Fed. Cir. 1983).

Even assuming *arguendo* that independent claims 1, 7, 14 and 22-23 are not patentable under 35 U.S.C. §103(a), which Appellants do not believe, dependent claims 2-6, 8-13, 15-21 and 24-28 are separately patentable from parent claims 1, 7, 14 and 22-23 for reasons presented hereinbelow.

For example, dependent claims 4, 15 and 24 further defines that the IOP comprises: “one or more IO processors, IO devices, a device driver module and a communication layer which defines a mechanism for communications between the Local Transport and the device driver module” which is **not** disclosed anywhere in Heil '374 or Angelo '794.

However, the Examiner asserts that the combined teachings of Heil '374 and Angelo '794 further teach,

wherein said IOP which comprises: at least one or more input/output processors (Heil: col 9/lines 49-59, element (114.9); at least one storage device as said input/output devices (col 6/lines 65-col 7/line 4, col 7/lines 51-56); a device driver module arranged to control access via interface means with said storage device (Heil: col 7/line 1-56); a communication layer which defines a

mechanism for communications between the Local Transport and the device driver module (Heil: col 10/lines 66-col 11/line 11, 45-53, element (240)).

This assertion is incorrect. Again, none of the cited portions of Heil '374 discloses what the Examiner has alleged. This is because no where in Heil '374 is there disclosure of any IOP as defined in claims 4, 15 and 24. Therefore, the cited portions of Heil '372 do NOT disclose the specific configuration of an IOP as comprising one or more IO processors, IO devices, a device driver module and a communication layer which defines a mechanism for communications between the Local Transport and the device driver module.

Dependent claims 5, 9, 16 and 25 further define that the "communication layer is responsible for managing all service requests and providing a set of Application Programming Interfaces (APIs) for delivering messages, along with a set of support routines that process the messages" which is **not** disclosed anywhere in Heil '374 or Angelo '794.

However, the Examiner asserts that the combined teachings of Heil '374 and Angelo '794 further teach,

wherein said communication layer is responsible for managing all service requests (Heil: col 11/lines 36-53, col. 12/lines 9-19) and providing a set of software comprising used to initiate communication with a network services via peer-to-peer communication or call services (i.e. APIs), (Heil: col 9/lines 11-18) for delivering messages, along with a set of support routines that process the messages (Heil: col 8/lines 24-49).

Again the assertion is incorrect. None of the cited portions of Heil '374 discloses what the Examiner has alleged. If Heil '374 does NOT disclose any IOP as defined in claims

4, 15 and 24, then there is no possibility of any disclosure from Heil '374 of any claimed "communication layer [of an IOP] for managing all service requests and providing a set of Application Programming Interfaces (APIs) for delivering messages, along with a set of support routines that process the messages" as defined in claims 5, 10, 16 and 25.

Dependent claims 6, 10, 16 and 25 further define that the "communication layer comprises a message layer which sets up a communication session, and a transport layer which defines how information will be shared" which is **not** disclosed anywhere in Heil '374 or Angelo '794.

However, the Examiner asserts that the combined teachings of Heil '374 and Angelo '794 further teach,

wherein said communication layer comprises a message layer which sets up a communication session (Heil: col 11/lines 66-col 11/line 11, 45-53), and a transport layer which defines how information will be shared (Heil: col 10/lines 66-col 11/line 11, coordinate retrieval of shared information).

Again, if Heil '374 does NOT disclose any IOP as defined in claim 4, then there is no possibility of any disclosure from Heil '374 of any claimed "communication layer [of an IOP] which sets up a communication session, and a transport layer which defines how information will be shared" as defined in claims 6, 10, 16 and 25.

Dependent claim 11 further defines that "the host driver module and the device driver module constitute a single device that is portable across a plurality of operating systems and host network platforms, and works interoperably with a plurality of storage devices and

operating systems" which is **not** disclosed anywhere in Heil '374 or Angelo '794.

However, the Examiner asserts that the combined teachings of Heil '374 and Angelo '794 further teach,

wherein said system driver module and said device driver module constitute a single device (Heil: col 4/lines 3-4) that is operable independent of the host operating system and operable at any different host computer (i.e. portable) across a plurality of operating systems and host network platforms (Heil: col 4/lines 13-29), and works interoperably with a plurality of storage devices and operating systems (Heil: col 4/lines 4-20).

Again, none of the cited portions of Heil '374 discloses what the Examiner has alleged. This is because no where in Heil '374 is there disclosure of any "Local Transport" and "Remote Transport" as defined in independent claim 9. Similarly, no where in Heil '374 is there disclosure of any combination of "system driver module" and "device driver module" as defined in independent claim 9. Therefore, the cited col. 4, lines 3-4 of Heil '372 simply describes the use of a host bus adapter (HBA).

Likewise, dependent claims 20 and 27 further define that "the Local Transport has a send handler function and the Remote Transport has a receive handler function which are respective program interfaces for receiving an inbound message from a remote server on a computer network for direct access to local input/output platform and for delivering an outbound message to said remote server on a computer network" which is **not** disclosed anywhere in Heil '374 or Angelo '794.

Nevertheless, the Examiner asserts that the combined teachings of Heil '374 and

219.36435X00
Serial No. 09/215,788

Angelo '794 further teach,

wherein said Local Transport further has a send means and said Remote Transport further has a receive means which are respective program interfaces for receiving an receiving message from a remote server on said computer network for direct access to local input/output platform and for delivering an sending message to said remote server on said computer network (Angelo: receiving means- col 2/lines 4-14, col 2/line 66-col 3/line 10, 41-48, sending means, col 3/lines 11-21).

Again, if Heil '374 and Angelo '794 do NOT disclose any "Local Transport" and "Remote Transport" as defined in claims 19 and 23, then there is no possibility of any disclosure from Heil '374 or Angelo '794 of any claimed "send handler function" and "receive handler function which are respective program interfaces for receiving an inbound message from a remote server on said computer network for direct access to local input/output platform and for delivering an outbound message to said remote server on said computer network" as expressly defined in claims 20 and 27.

In view of the foregoing explanations, and in view of the fact that neither Heil '374 nor Angelo '794, whether taken in combination or individually, discloses and suggests Appellants' dependent claims 2-6, 8-13, 15-21 and 24-28, Appellants respectfully request that the rejection of dependent claims 2-6, 8-13, 15-21 and 24-28 be reversed as well.

IX. Conclusion

In view of the law and the facts presented herein, Appellants respectfully submit that

the Examiner's proposed combination of Heil '374 and Angelo '794 fails to disclose or suggest Appellants' claimed invention. The §103 art rejection is essentially based on the erroneous belief that Heil '374 and Angelo '794 discloses all the claimed features. However, the Appellants have established this as incorrect. None of prior art references was correctly relied by the Examiner to support the §103 rejections. As stated in In re Schaefer, 229 F.2d 476, 108 U.S.P.Q. 326 (C.C.P.A 1956):

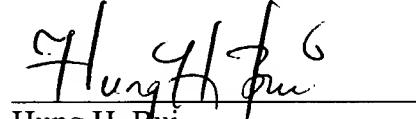
"[T]o determine whether the combination of references is improper, the following criterion is often used: namely, whether the prior art suggests doing what applicant has done ... [I]t is not enough for a valid rejection to review the prior art in retrospect once an applicant's disclosure is known. The art applied should be viewed by itself to see if it fairly disclosed doing what an applicant has done." *See also In re Taborski*, 556 F.2d 85, 183 USPQ 350 (CCPA 1973); *In re Regal*, 526 F.2d 1399, 188 USPQ 136 (CCPA 1976); and *In re Imperato*, 46 F.2d 585, 179 USPQ 730 (CCPA 1973).

Here, it is abundantly clear that all cited prior art references simply do not disclose or suggest what Appellants have done. For all of the foregoing reasons, Appellants respectfully request the Board reverse the final rejection of claims 1-28.

219.36435X00
Serial No. 09/215,788

A fee of \$320.00 is incurred by the Appeal Brief. Please charge any shortage of fees due in connection with the filing of this paper, including extension of time fees, to the Deposit Account of Antonelli, Terry, Stout & Kraus, No. 01-2135 (Application No. 219.36435X00), and please credit any excess fees to said deposit account.

Respectfully submitted,
Antonelli, Terry, Stout & Kraus, L.L.P.



Hung H. Bui
Attorney for the Appellants
Registration No.: 40,415

Date: **November 13, 2001**
Antonelli, Terry, Stout & Kraus, LLP
1300 North Seventeenth Street
Suite 1800
Arlington, VA 22209
(703) 312-6600 (phone)
(703) 312-6666 (fax)

Appendix:

Claims 1-28 On Appeal

1 1. (Amended) An input/output platform (IOP) access module for providing
2 input/output device access between a host system and another system, via a data network,
3 said IOP access module comprising:

4 a Local Transport arranged to provide an interface to an input/output platform (IOP)
5 supporting an array of input/output devices;

6 a Remote Transport arranged to provide an interface to said another system, via said
7 data network; and

8 a Connection Manager arranged to establish connection services and to create a direct
9 call path between the Local Transport and the Remote Transport so as to provide access to
10 input/output devices.

1 2. The input/output platform (IOP) access module of claim 1, wherein said IOP
2 access module is one of a hardware module, a combined hardware/software module, and a
3 software module provided on a tangible medium.

1 3. (Amended) The input/output platform (IOP) access module of claim 1,

2 wherein said host system corresponds to a host server, said another system corresponds to any
3 one of remote servers, via said data network.

1 4. The input/output platform (IOP) access module of claim 2, further comprising
2 said IOP which comprises:
3 at least one or more input/output processors;
4 at least one storage device as said input/output devices;
5 a device driver module arranged to interface with said storage device;
6 a communication layer which defines a mechanism for communications between the
7 Local Transport and the device driver module.

1 5. The input/output platform (IOP) access module of claim 4, wherein said
2 communication layer is responsible for managing all service requests and providing a set of
3 Application Programming Interfaces (APIs) for delivering messages, along with a set of
4 support routines that process the messages.

1 6. The input/output platform (IOP) access module of claim 5, wherein said
2 communication layer comprises a message layer which sets up a communication session, and
3 a transport layer which defines how information will be shared.

1 7. (Amended) A host system, comprising:
2 a processor;
3 an array of storage devices;
4 a driver module for exporting local storage device access onto a computer network,
5 said driver module comprising:
6 a device driver module arranged to provide an interface to said array of local
7 storage devices;
8 a host driver module arranged to provide an interface to an operating system,
9 said host driver module comprising a Local Transport which communicates with the
10 device driver module, a Remote Transport which provides an interface to said
11 computer network, and a Connection Manager which establishes connection services
12 with remote systems on said computer network and coordinates functions responsible
13 for creating a direct call path between the Local Transport and the Remote Transport
14 to provide access to said storage devices; and
15 a communication layer which supports communications between the host
16 driver module and the device driver module.

1 8. The host system of claim 7, wherein said host system corresponds to a host
2 server, said remote systems corresponds to remote servers arranged in a cluster, and said
3 computer network corresponds to a system area network for communications between said

4 host system and said remote systems within said cluster.

1 9. The host system of claim 7, wherein said communication layer is responsible
2 for managing all service requests and providing a set of Application Programming Interfaces
3 (APIs) for delivering messages, along with a set of support routines that process the
4 messages.

1 10. The host system of claim 9, wherein said communication layer comprises a
2 message layer which sets up a communication session, and a transport layer which defines
3 how information will be shared.

1 11. (Amended) The host system of claim 9, wherein said host driver module
2 and said device driver module constitute a single device that is portable across a plurality of
3 operating systems and host network platforms, and works interoperably with a plurality of
4 storage devices and operating systems.

1 12. (Amended) The host system of claim 9, wherein said host driver module
2 and said device driver module operate in accordance with an Intelligent Input/Output (I₂O)
3 specification for allowing storage devices to operate independently from the operating
4 system.

1 13. The host system of claim 7, wherein said driver module is one of a hardware
2 module, a combined hardware/software module, and a software module provided on a
3 tangible medium.

1 14. A driver configuration of a host server for exporting local storage device
2 access onto a computer network, comprising:
3 an input/output platform (IOP) arranged to control an array of local storage devices;
4 and
5 a system driver module comprising:
6 a Local Transport arranged to provide an interface to said input/output
7 platform (IOP);
8 a Remote Transport arranged to provide an interface to said computer
9 network; and
10 a Connection Manager arranged to establish connection services with
11 remote servers on said computer network and coordinate functions responsible
12 for creating a direct call path between the Local Transport and the Remote
13 Transport to provide access to the local storage devices.

1 15. The driver configuration of claim 14, wherein said input/output platform (IOP)

2 supports at least one or more input/output processors, and comprises:
3 a device driver module which interfaces the local storage devices for controlling said
4 array of local storage devices; and
5 a communication layer which defines a mechanism for communications between the
6 system driver module and the device driver module.

1 16. The driver configuration of claim 15, wherein said communication layer is
2 responsible for managing and dispatching all service requests and providing a set of
3 Application Programming Interfaces (APIs) for delivering messages, along with a set of
4 support routines that process the messages, and is comprised of a message layer which sets up
5 a communication session, and a transport layer which defines how information will be shared.

1 17. The driver configuration of claim 15, wherein said system driver module and
2 said device driver module constitute a single device that is portable across a plurality of host
3 operating systems and host network platforms, and works interoperably with a plurality of
4 storage devices and host operating systems.

1 18. The driver configuration of claim 15, wherein said system driver module and
2 said device driver module operate in accordance with an Intelligent Input/Output (I₂O)
3 specification for allowing storage devices to operate independently from the operating system

4 of the host server.

1 19. The driver configuration of claim 15, wherein, upon initialization, said Local
2 Transport scans the local bus so as to locate and initialize all local input/output platforms
3 (IOPs) and builds an opaque "context" structure for each input/output platform (IOP),
4 wherein said Remote Transport prepares to accept requests from a remote server through said
5 computer network, and wherein said Connection Manager queries said Local Transport so as
6 to determine the number of input/output platforms (IOPs), builds an IOP descriptor structure
7 for each input/output platform (IOP) which includes an exported table of function call
8 pointers and the context required by the Local Transport to communicate with the
9 input/output platform (IOP), and finally establishes a network management communication
10 channel through the Remote Transport, which waits for an external connection from said
11 remote server on said computer network for exporting local device access onto said computer
12 network using said direct call path between the Local Transport and the Remote Transport.

1 20. The driver configuration of claim 19, wherein said Local Transport further has
2 a send handler function and said Remote Transport further has a receive handler function
3 which are respective program interfaces for receiving an inbound message from a remote
4 server on said computer network for direct access to local input/output platform and for
5 delivering an outbound message to said remote server on said computer network.

1 21. The driver configuration of claim 19, wherein said Remote Transport further
2 builds an IOP connection structure including at least an IOP descriptor pointer which refers to
3 the IOP descriptor structure of the Connection Manager for making a direct call to the Local
4 Transport through the receive handler function and the send handler function.

1 22. A process of exporting storage device access onto a computer network using
2 an input/output platform (IOP) access module of a host server, comprising the steps of:
3 providing an interface to an input/output platform (IOP) supporting an array of
4 storage devices;
5 providing an interface to a remote server on said computer network;
6 establishing service connection between said host server and said remote server on
7 said computer network in response to a request from a remote server on said computer
8 network; and
9 providing a direct call access to said storage devices for said remote server to share
10 resources of said storage devices while bypassing operating system protocol stacks.

1 23. (Amended) A process of establishing a service connection to a local
2 input/output platform (IOP) connected to a local bus using a system driver module in
3 response to a request from a remote server on a system area network, comprising the steps of:

4 beginning initialization of said driver module which provides access to a local storage
5 system while bypassing protocol stacks of a host operating system, said system driver module
6 comprising a Local Transport which provides direct access to the local storage device system,
7 a Remote Transport which interfaces to other nodes of said system area network, and a
8 Connection Manager which provides connection services and coordinates functions
9 responsible for creating a direct call path between the Local Transport and the Remote
10 Transport;

11 scanning, at said Local Transport, the local bus to locate and initialize all local
12 input/output platforms (IOPs), and building an IOP context structure for each input/output
13 platform (IOP) found;

14 preparing, at said Remote Transport, to accept a request for a service connection from
15 said remote server on said system area network;

16 asking, at said Connection Manager, whether said Local Transport determines the
17 number of input/output platforms (IOPs), and building a descriptor structure for each
18 input/output platform (IOP) which includes an exported table of function call pointers and the
19 context required by the Local Transport to communicate with the input/output platform
20 (IOP); and

21 establishing a system area network management communication channel through the
22 Remote Transport, which waits for an external connection from said remote server on said
23 system area network for exporting local device access onto said system area network using

24 said direct call path between the Local Transport and the Remote Transport.

1 24. The process of claim 23, wherein said input/output platform (IOP) comprises:
2 a device driver module which interfaces the local storage devices, and which controls
3 an array of local storage devices; and
4 a communication layer which defines a mechanism for communication between the
5 system driver module and the device driver module.

1 25. The process of claim 24, wherein said communication layer is responsible for
2 managing and dispatching all service requests and providing a set of Application
3 Programming Interfaces (APIs) for delivering messages, along with a set of support routines
4 that process the messages, and is comprised of a message layer which sets up a
5 communication session, and a transport layer which defines how information will be shared.

1 26. The process of claim 23, wherein said system driver module and said device
2 driver module constitute a single device that is portable across a plurality of host operating
3 systems and host network platforms, and operate in accordance with an Intelligent
4 Input/Output (I₂O) specification for allowing storage devices to operate independently from
5 the host operating system.

1 27. The process of claim 23, wherein said Local Transport further has a send
2 handler function and said Remote Transport further has a receive handler function which are
3 respective program interfaces for receiving an inbound message from a remote server on said
4 system area network for direct access to local input/output platform (IOP) and for delivering
5 an outbound message to said remote server on said system area network.

1 28. The process of claim 27, wherein said Remote Transport further builds an IOP
2 connection structure including at least an IOP descriptor pointer which refers to the IOP
3 descriptor structure of the Connection Manager for making a direct call to the Local
4 Transport through the receive handler function and the send handler function.